REMARKS

In view of the following discussion, the Applicants submit that none of the claims now pending in the application is obvious under the provisions of 35 U.S.C. §103. Thus, the Applicants believe that all of these claims are in allowable form.

I. REJECTION OF CLAIMS 1, 2, 7, 8, 13, 14, 20, 21, 24, 25, 28, AND 29 UNDER 35 U.S.C. § 103

Claims 1, 2, 7, 8, 13, 14, 20, 21, 24, 25, 28, and 29 stand rejected as being unpatentable over the Nine et al. patent (U.S. 6,560,611, issued May 6, 2003, hereinafter "Nine"). The Applicants respectfully traverse the rejection. Specifically, the Applicants submit that Nine fails to teach, show, or suggest several of the limitations recited in Applicants' independent claims 1, 7, 13, 20, 24, and 28.

Primarily, the Applicants submit that Nine is completely devoid of any teaching, showing, or suggestion relating to the comparison of an alert (indicating an attack or anomalous incident) – or more specifically, the comparison of features of the alert - to the features of existing alert classes, in order to classify the alert, as claimed by the Applicants in independent claims 1, 7, 13, 20, 24, and 28.

By contrast, Nine teaches a network monitoring system that simply reports a detected problem to the proper individual (*e.g.*, technician), based on the nature of the problem. That is, Nine does not classify the detected problem (*e.g.*, in accordance with its features) by comparing it to known problems, but simply evaluates the detected problem as a discrete incident and reports it to a human technician for further action.

Specifically, Nine teaches a remote monitoring system (RMS) that reports to a network operation site (NOS) when the RMS detects an anomaly with respect to a service it monitors. The report provided by the RMS is a ticket or data record containing information about the service (*e.g.*, location, severity of problem, time of occurrence). In addition, the system "determines the nature of the problem, and notifies the proper personnel [*e.g.*, a technician]" (*See*, Nine at column 3, lines 25-27).

For instance, the portion of Nine that the Examiner cites to teach the limitations of "updating a minimum similarity expectation for one or more features" of an alert relative to features of alert classes and of "updating a similarity expectation for one or

more features" of an alert relative to features of alert classes in fact merely teaches that the monitoring software is replicated for each service on a device by an informer engine executing forker software and sender software (*See, e.g.* Nine at column 5, line 45 – column 6, line 9). There is no discussion of examining the features of an alert, or of the need to update a minimum similarity requirement or a similarity expectation for the features of the alert to the one or more alert classes.

The portion of Nine that the Examiner cites to teach the limitations of "comparing [a] new alert with one or more alert classes" and "associating the new alert with the existing alert class that the new alert most closely matches" in fact merely teaches three techniques for detecting a problem with a monitored service. The first technique checks to make sure that the service is responsive (*e.g.*, by "ping, nmap, finger, or telnet", Nine at column 7, lines 25-33). The second technique monitors environmental sensors to detect problems with the environment (*e.g.*, "if the temperature is too high", Nine at column 7, lines 34-39). The third technique examines a log of the monitored service and parses for potential problems (*e.g.*, indication that a particular route associated with a router is not functioning, Nine at column 7, lines 40-46). None of these techniques involve the comparison of an alert to existing alert classes, or the association of the alert with one of the existing alert classes based on the comparison.

The portion of Nine that the Examiner cites to teach the limitations of "comparing [a] new alert with one or more alert classes" and "defining a new alert class that is associated with the new alert" in fact merely teaches that log files for a monitored service may be used to diagnose problems with the service. Again, there is no mention of the need to compare an alert with existing alert classes in order to classify the alert, as claimed by the Applicants.

Moreover, Nine does not even teach, show, or suggest that an alert may be classified in accordance with its features. The portion of Nine that the Examiner cites to teach the limitation of "identifying a set of potentially similar features shared by [a] new alert and one or more existing alert classes" in actuality merely teaches that software monitors a service and reports to the NOS when the service is unresponsive or when an anomaly is detected. The report contains "information about the service, such as location, severity of the problem, and time of occurrence" (*See, e.g.*, Nine at column 3, lines 12-20). There is no mention in this passage of the need to identify features of the

problem or to compare the problem to other known problems (*e.g.*, existing alert classes) based on the identified features.

In short, as discussed above, Nine fails to teach, show, or suggest any sort of classification of alerts by comparing features of the alerts to features of existing alert classes, as recited by the Applicants in independent claims 1, 7, 13, 20, 24, and 28. Specifically, Applicants' claims 1, 7, 13, 20, 24, and 28, as amended, positively recite:

1. In an intrusion detection system that includes a plurality of sensors that generate alerts when attacks or anomalous incidents are detected, a method for organizing alerts into alert classes, both the alerts and alert classes having a plurality of features, the method comprising the steps of:

(a) receiving a new alert;

(b) identifying a set of potentially similar features shared by the new alert and one or more existing alert classes;

(c) updating a minimum similarity requirement for one or more features;

(d) updating a similarity expectation for one or more features;

(e) comparing the new alert with one or more alert classes, and either:

(f1) associating the new alert with the existing alert class that the new alert most closely matches; or

(f2) defining a new alert class that is associated with the new alert. (Emphasis added)

7. A computer readable medium containing an executable program for organizing alerts that are generated by a plurality of sensors into alert classes, both the alerts and alert classes having a plurality of features, where the program performs the steps of:

(a) receiving a new alert;

(b) identifying a set of potentially similar features shared by the new alert and one or more existing alert classes;

(c) updating a minimum similarity requirement for one or more features;

(d) updating a similarity expectation for one or more features;

(e) comparing the new alert with one or more alert classes, and either:

(f1) associating the new alert with the existing alert class that the new alert most closely matches; or

(f2) defining a new alert class that is associated with the new alert. (Emphasis added)

13. In an intrusion detection system that includes a plurality of sensors that generate alerts when attacks or anomalous incidents are detected, a system for organizing alerts into alert classes, both the alerts and alert classes having a plurality of features, where the system comprises:

(a) means for receiving a new alert;

(b) means for identifying a set of potentially similar features shared by the new alert and one or more existing alert classes;

(c) means for updating a minimum similarity requirement for one or more features;

(d) means for updating a similarity expectation for one or more features;

(e) means for comparing the new alert with one or more alert classes; and

(f1) means for associating the new alert with the existing alert class that the new alert most closely matches, or defining a new alert class that is associated with the new alert. (Emphasis added)

20. A method for organizing alerts into alert classes, both the alerts and alert classes having a plurality of features, each feature having one or more values, the method comprising the steps of:

(a) receiving a new alert;

(b) identifying a set of potentially similar features shared by the new alert and one or more existing alert classes;

(c) updating a similarity expectation for one or more feature values;

(d) comparing the new alert with one or more alert classes, and either:

(e1) associating the new alert with the existing alert class that the new alert most closely matches; or

(e2) defining a new alert class that is associated with the new alert. (Emphasis added)

24. A computer readable medium containing an executable program for organizing alerts into alert classes, both the alerts and alert classes having a plurality of features, each feature having one or more values, where the program performs the steps of:

(a) receiving a new alert;

(b) identifying a set of potentially similar features shared by the new alert and one or more existing alert classes;

(c) updating a similarity expectation for one or more feature values;

(d) comparing the new alert with one or more alert classes, and either:

(e1) associating the new alert with the existing alert class that the new alert most closely matches; or

(e2) defining a new alert class that is associated with the new alert. (Emphasis added)

28. A system for organizing alerts into alert classes, both the alerts and alert classes having a plurality of features, each feature having one or more values, the system comprising:

(a) means for receiving a new alert;

(b) means for identifying a set of potentially similar features shared by the new alert and one or more existing alert classes;

(c) means for updating a similarity expectation for one or more feature values;

(d) means for comparing the new alert with one or more alert classes; and

(e1) means for associating the new alert with the existing alert class that the new alert most closely matches, or defining a new alert class that is associated with the new alert. (Emphasis added)

As discussed above, nowhere does Nine teach or even suggest the desirability of classifying of alerts by comparing features of the alerts to features of existing alert classes. Therefore, the Applicants submit that independent claims 1, 7, 13, 20, 24, and 28 fully satisfy the requirements of 35 U.S.C. §103 and are patentable thereunder.

Dependent claims 2, 8, 14, 21, 25 and 29 depend from claims 1, 7, 13, 20, 24, and 28 and recite additional features therefore. As such, and for at least the same reasons set forth above, the Applicants submit that claims 2, 8, 14, 21, 25 and 29 are not made obvious by the teachings of Nine. Therefore, the Applicants submit that dependent claims 2, 8, 14, 21, 25 and 29 also fully satisfy the requirements of 35 U.S.C. §103 and are patentable thereunder.

II. CONCLUSION

Thus, the Applicants submit that all of the presented claims fully satisfy the requirements of 35 U.S.C. §103. Consequently, the Applicants believe that all of these claims are presently in condition for allowance. Accordingly, both reconsideration of this application and its swift passage to issue are earnestly solicited.

If, however, the Examiner believes that there are any unresolved issues requiring the issuance of a final action in any of the claims now pending in the application, it is requested that the Examiner telephone Mr. Kin-Wah Tong, Esq. at (732) 530-9404 so that appropriate arrangements can be made for resolving such issues as expeditiously as possible.

Respectfully submitted,

8/23/07
Date

Kin-Wah Tong, Attorney
Reg. No. 39,400
(732) 530-9404

Patterson & Sheridan, LLP
595 Shrewsbury Avenue
Shrewsbury, New Jersey 07702